1 Forelesning 2

1.1 Hva er en datastruktur

En algoritme tar input, gjør beregninger, og gir output. Datastrukturer lagrer data og har flere forskjellige oppgaver som kan utføres på disse dataene.

> **Example. GPS Navigasjonsenhet**
>
> - Har veinettverk
> - Kan be om korteste vei fra start til mål
>
> Er dette en algoritme eller en datastruktur? ◇

`Collection <E> interface` er en samling med objekter av typen `E`. Viktige metoder er

- `size()`
- `contains(Object 0)`
- `add (E e)`
- `Remove(Object obj)`
- `toArray()`

`List` er en utvidelse av `Collection` med litt flere metoder. Elementene i en liste har en indeks, og noen viktige metoder i `List` er

- `indexOf(Object obj)`
- `get(int index)`
- `set(int index, E e)`

1.2 Array List

I `ArrayList` brukes en array av typen `Object[]`. Bare en del av arrayen har data. Når dette arrayet blir fullt må vi lage et større array. Dette betyr at det er lett å få `IndexOutOfBoundsException`. For eksempel kan man lage et nytt array av dobbel størrelse, hvor alle elementer fra forrige array kopieres over til det nye arrayet.

1.3 Linked List

En Linked List er en liste hvor hvert element i listen lagres i et eget node object. Hver node vet kun neste og forrige node. Listen vet kun første og siste node. For å finne node i, så starter vi på første node og "hopper" i ganger.

En linked list som kun vet neste element kalles en *Single Linked List*, og en linked list som vet både neste og forrige element kalles en *Double Linked List*.

1.4 Hvordan finne kjøretid på metoder?

Man må vite hvordan `ArrayList` og `LinkedList` er implementert. Ved å forstå hva `ArrayList` og `LinkedList` gjør, er det lett å forstå hva kjøretiden er. Ukesoppgaven er og implementere enkle versjoner av disse listene.

1.5 Kø og Stabel

Kø	Stabel
FIFO	FILO / LIFO
Legger til sist	Legger til sist
Fjerner først	Fjerner sist

1.5.1 Metoder i Queue og Stack

List	Queue	Stack
add(E e)	offer(E e)	push(E e)
remove(int i)	poll()	pop()
get(int i)	peek()	peek()

1.6 Set

Set er en Collection der hvert element kun kan være der en gang. Elementene har ikke en ebstemt ordning. Set har heller ikke `indexOf(element)` metoden.

	HashSet	TreeSet
`add()`	$O(1)^\star$	$O(\log n)$
`remove()`	$O(1)$	$O(\log n)$
`contains(obj)`	$O(1)$	$O(\log n)$

$$\int_{-\infty}^{\infty} \frac{1}{2}^2 dx = \nabla \times \vec{F} = \left| \begin{bmatrix} \vec{i} & \vec{j} & \vec{k} \\ \partial_x & \partial_y & \partial_z \\ \end{bmatrix} \right| = \vec{i}\,(\partial_y - \partial_z) - \vec{j}\,(\partial_x - \partial_z) + \vec{k}\,(\partial_x - \partial_y)$$